# Discretization of the Solution Domain

*Pertinent reading:* *Section 8.2* in textbook

Department of Civil and Environmental Engineering

*Presented by:*

V.N. Kaliakin

University of Delaware



## General Steps in Performing Finite Element Analyses

- Discretization of the solution domain.
- Assembly of *element* equations to form the *global* equations.
- Application of *nodal specifications*.
- Solution of the *global* equations.
- Calculation of the *secondary dependent variables* (e.g., "gradients").
- Post-processing of the results.
- Interpretation of the results.



## Contents of Presentation

- General Remarks
- Domain Discretization Error
- Common Element Types
- Element Characteristics
- Placement of Elements
- Element Shapes
- Mesh Generation



## General Remarks

- The goal of the first task is to discretize the domain ∧ *as accurately as possible*.
- This is realized by breaking up ∧ into a *finite* number of *non-overlapping* elements.
- This process is commonly called meshing; the collection of elements is referred to as a mesh.
- A high-quality mesh is requisite to convergent and accurate FE solutions.



## Contents of Presentation

- General Remarks
- Domain Discretization Error
- Common Element Types
- Element Characteristics
- Placement of Elements
- Element Shapes
- Mesh Generation



## Domain Discretization Error

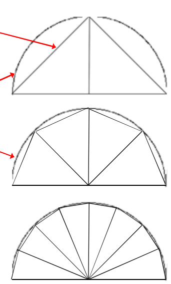
Recall (from Chapter 6):

- The meshing of two- and three-dimensional domains has the potential of introducing domain discretization error into the analysis.
- This geometric error arises when the elements along the boundary do not exactly fit the geometry of the actual body.



Finite element domain

Actual domain

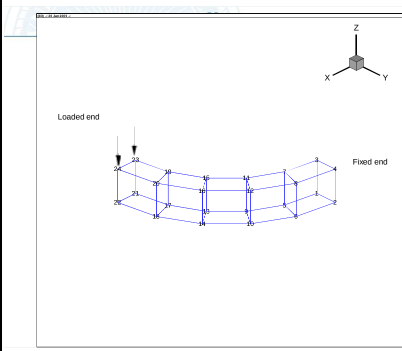




## Domain Discretization Error

- Domain discretization errors arise when non-polynomial boundaries are approximated by polynomial elements.





## Domain Discretization Error

- Depending on the types of elements used and on the placement of the associated nodes, this error manifests itself in small portions of the domain not being covered by an element, or extra portions of the mesh "spilling over" the boundaries of the domain.



## Domain Discretization Error

- In either case, the approximate solution domain *differs* from the actual one and some amount of additional *error* is introduced into the solution.
- Additional details pertaining to domain discretization error are given in *Section 8.2.1* of the textbook.



## Contents of Presentation





## Common Element Types

- In theory, a variety of element shapes could be used in a given mesh.
- However, in practice only a *few* different shapes are typically employed.
- Some examples of commonly used one-, two-and three-dimensional elements are now considered (Chapter 9 presents additional details).



## Common Element Types

General Remarks:

- The designations "one-dimensional", "two-dimensional" and "three-dimensional" refer only to the *spatial extent* of the element domain.
- They have nothing to do with the number of nodes per element, unknowns per node, etc.

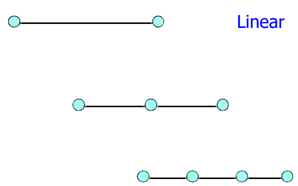


## One-Dimensional Elements

- The discretization of a one-dimensional domain is quite straightforward, and introduces *no discretization error*.
- Element interfaces are simply (common) nodes.
- One-dimensional (line) elements are quite simple in nature.



## One-Dimensional Elements



Linear

Quadratic

Cubic



## Two-Dimensional Elements

- In theory, when discretizing a two-dimensional domain, the number of element types available is *infinite*.
- Element interfaces are *common edges*.
- In practice combinations of a few *simple element shapes* typically prove adequate.

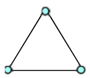


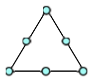
## Two-Dimensional Elements

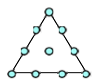
- The simplest polygonal shape that will span a two-dimensional domain is the *triangle*.
- Triangular elements are thus sometimes referred to as the two-dimensional "simplex" elements.

## Two-Dimensional Elements

Triangular Family of Elements

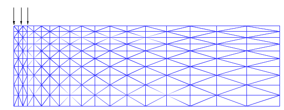

Linear


Quadratic


Cubic



## Application of Triangular Elements: Numerical Simulation of a Soil Layer Underlain by a Rigid Base and Loaded by Infinite Strip Footing

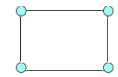


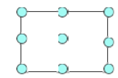


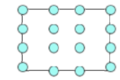
## Two-Dimensional Elements

*Lagrange* Family of Elements

(to be discussed in Chapter 9)


Linear


Quadratic

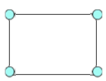

Cubic

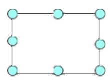


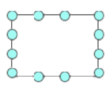
## Two-Dimensional Elements

*Serendipity* Family of Elements

(to be discussed in Chapter 9)

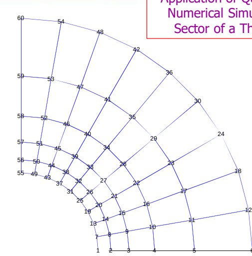

Linear


Quadratic


Cubic



## Application of Quadrilateral Elements: Numerical Simulation of 90-Degree Sector of a Thick-Walled Cylinder





## Three-Dimensional Elements

- The discretization of three-dimensional domains is *more complex* than two-dimensional ones.
- Element interfaces are *element faces*.
- In practice combinations of a *few simple element shapes* again typically prove adequate.



## Three-Dimensional Elements

- The simplest polygonal shape that will span a three-dimensional domain is the *tetrahedron*.
- Tetrahedral elements are thus sometimes referred to as the three-dimensional “simplex” elements.

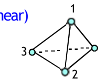


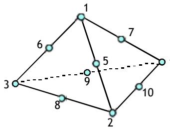
## Three-Dimensional Elements

*Tetrahedral* Family of Elements

(to be discussed in Chapter 9)

4-node (linear)



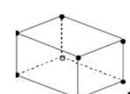

10-node (quadratic)



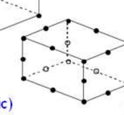
## Three-Dimensional Elements

Hexahedral (“Brick”) Family of Elements

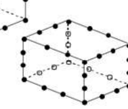
8-node (linear)



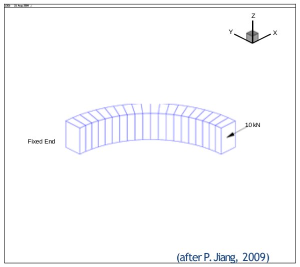
20-node (quadratic)



32-node (cubic)



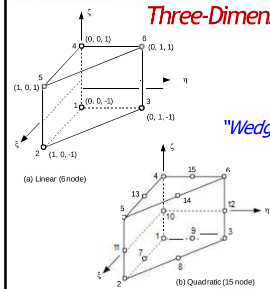
## Application of Hexahedral Elements: Numerical Simulation of a Curved Beam




(after P. Jiang, 2009)



## Three-Dimensional Elements

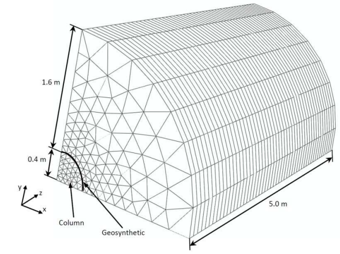



*"Wedge"* Family of Elements

(to be discussed in Chapter 9)



## Application of Wedge Elements: Numerical Simulation of a Stone Column




(after Khabbazian, Meehan & Kaliakin 2009)



## Contents of Presentation





## Element Characteristics

- Shape (e.g., a *line*, *triangle*, *quadrilateral*, *tetrahedron*, *hexahedron*, etc.).
- *Number of nodes* in the element and the global node numbers assigned to these nodes.
- Type of nodes present in the element. Nodes are either *interior* or *exterior* to the element. Interior nodes are those nodes not shared by neighboring elements. Exterior nodes are either corner nodes located at the *vertices*, or are those located along an *element edge*.

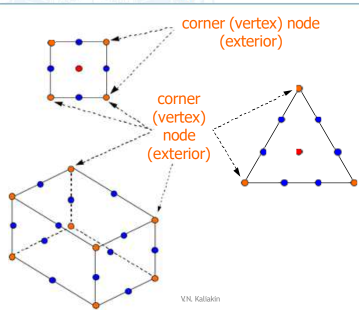


## Element Characteristics

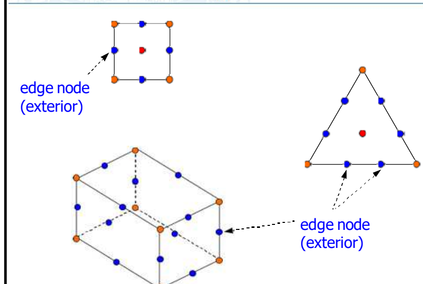
Type of nodes present in the element.

- Nodes are either ***interior*** or ***exterior*** to the element.
- Interior nodes are those nodes ***not*** shared by neighboring elements.
- Exterior nodes are either corner nodes located at the ***vertices***, or are those located along an ***element edge***.

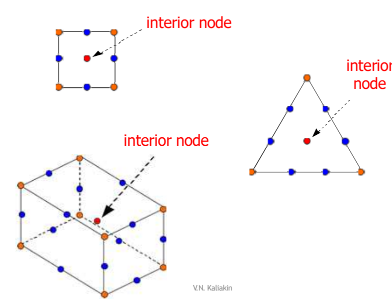








## Element Characteristics

- Type(s) of nodal degrees of freedom (recall Step 1 in Chapter 7).
- Order of interpolation functions used to describe: a) the element geometry and, b) the approximation of the nodal values of the primary dependent variables.
- Parameters describing the material constitution of the continuum being discretized by the element (recall Step 1 in Chapter 7).



## Contents of Presentation





## Placement of Elements

- The *accuracy* of FE analyses is largely dependent upon the type (order) of element used and upon the distribution of these elements.
- In discretizing a domain, the distribution and location of elements is based largely on experience of the analyst.
- Usually one starts with a simple mesh and subsequently *refines* the mesh to a finer one.



## Placement of Elements

Some Guidelines:

- Element boundaries should be placed along all exterior and interior boundaries of the body being discretized, in a manner that *minimizes* the domain discretization error.



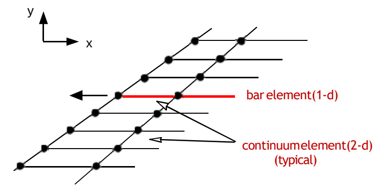
## Placement of Elements

- Since material and geometric parameters are typically assumed to be constant within an element, element interfaces must coincide with material interfaces.



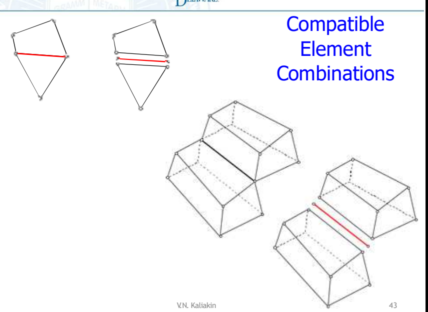
## Placement of Elements

**Example:** modeling embedded inclusions






Compatible Element Combinations





## Placement of Elements

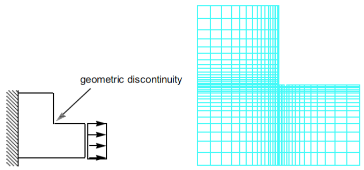
- Nodes should be placed at any location in ∧ a which a value of the primary dependent variable(s) is desired.



## Placement of Elements

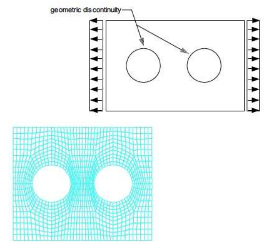
- Even in simple meshes, it is common practice to place the bulk of the elements in locations where the *gradient* of the primary dependent variables would be expected (or is known) to be high.

## Placement of Elements



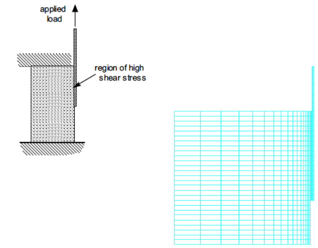



## Placement of Elements






## Placement of Elements






## Placement of Elements

- Elements should *not be overly distorted.*
- Although further details pertaining to this issue will be given in Chapters 10 & 13, some general comments follow.



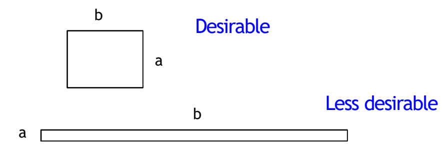
## Contents of Presentation





## Element Shapes

- In quadrilateral elements, the *aspect ratio* ($a/b$ or $b/a$) should be kept near unity.






## Element Shapes

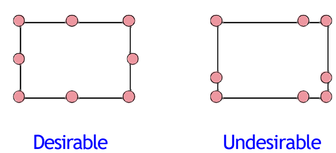
- The vertex angles in *quadrilateral* elements should, ideally, be near 90 degrees.


- In *triangular* elements these angles should, ideally, be near 60 degrees.




## Element Shapes

- Edge nodes should be uniformly distributed.






## Element Shapes

- Additional details pertaining to acceptable element shapes are given in Chapters 10 & 13 of the textbook.

## Contents of Presentation





## Mesh Generation

- The generation of high quality meshes has been a very active area of research.
- Driving this research is the realization that possibly the most important task to automate in the entire finite element simulation process is the generation of high quality meshes.



## Mesh Generation

Mesh generation is complicated by the realization that a mesh must satisfy *almost contradictory* requirements:

- It must conform to the shape of the simulated domain ∧; it must be composed of elements that are the "right" size and shape, avoiding excessive distortion (e.g., overly small or overly large interior angles).
- Over relatively short distances it will typically have to grade from large elements (coarse discretization) to small elements (fine discretization).

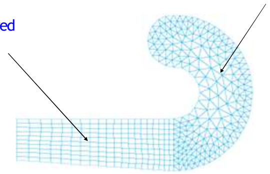


## Mesh Generation

- Mesh types fall under two main headings, namely ***structured*** and ***unstructured***.
- A mesh is called ***structured*** if its connectivity is of the "finite difference" type.
- If its connectivity is of any other type, the mesh is called ***unstructured***.

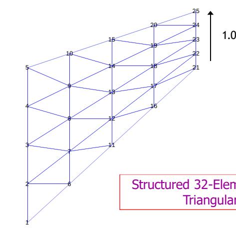


## Mesh Generation





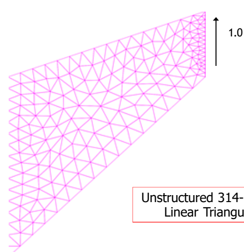

Inclined Beam Subjected to Tangential Traction




Structured 32-Element Mesh of Linear Triangular Elements



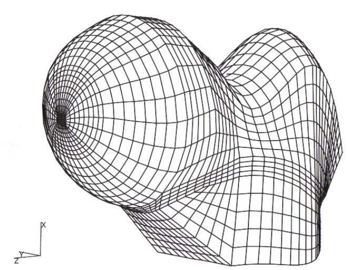
Inclined Beam Subjected to Tangential Traction




Unstructured 314-Element Mesh of Linear Triangular Elements

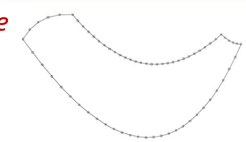


Unstructured mesh of 8-node hexahedral elements

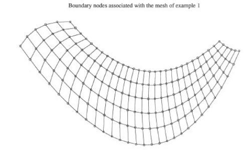




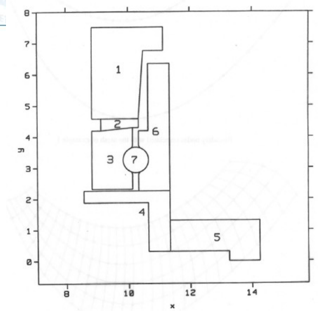

## Meshing Example



Boundary nodes associated with the mesh of example 1

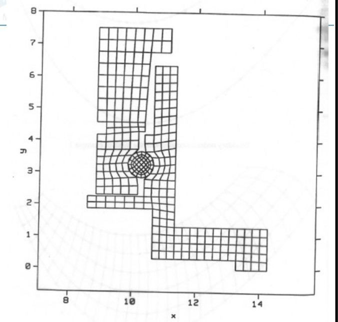


*Meshing Example*



V.N. Kaliakin 64

*Meshing Example*



*Meshing Example*